AD-A119 156 MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB F/G 9/2
NEW CONSOLE INTERFACE SYSTEM.(U)
JUL 82 A J WARDROP F19628-80-C-0002
UNCLASSIFIED ETS-65 ESD-TR-82-048 NL

1 OF 1
AD
A119156

END
DATE
FILMED
10-82
DTIC

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

# NEW CONSOLE INTERFACE SYSTEM

*A.J. WARDROP*

*Group 94*

**PROJECT REPORT ETS-65**

**14 JULY 1982**

**Approved for public release; distribution unlimited.**

A

LEXINGTON MASSACHUSETTS

## ABSTRACT

A new console interface system has been designed and built for the Experimental Test Site (ETS) of the Ground Electro-Optical Deep Space Surveillance (GEODSS) program using microprocessor technology. The new system allows current and future microprocessor based subsystems to be conveniently integrated into the overall system. Serial interfaces, high performance, multi-use DMA interfaces, and shared memory systems are extensively used. The use of LSI, standard microcomputer boards, and built-in tests and diagnostics is expected to result in a more reliable and maintainable system.

CONTENTS

## I. INTRODUCTION

A new console interface system has been designed and built for the Experimental Test Site (ETS) of the Ground Electro-Optical Deep Space Surveillance (GEODSS) program using microprocessor technology. The new system allows current and future microprocessor based subsystems to be conveniently integrated into the overall system. Most of the low level hardware control previously done by the central Modcomp minicomputers is now accomplished by distributed intelligence in the hardware modules, reducing Modcomp overhead. The use of serial interfaces and high performance, multi-use DMA and shared memory access interfaces yields a greater data transfer capacity with much less cabling than the current system. The use of LSI, standard microcomputer boards, and built-in tests and diagnostics is expected to result in a more reliable and maintainable system.

## II. SYSTEM OVERVIEW

### A. System Description

The basic telescope system consists of a telescope, sensor package, Modcomp IV-25 minicomputer, and console. The Modcomp drives the telescope, takes data, performs satellite file maintenance, generates search patterns, and runs calibration procedures. The operator directs the Modcomp through an array of pushbuttons, a joystick, and a CRT terminal. A cursor slaved to the joystick as an overlay on the television signal from the main optical sensor allows the operator to indicate relative positions to the Modcomp, and therefore bring an object to telescope boresight. The Modcomp telescope operating program is described in a separate publication (ETS-56).

The console interface system consists of a console processor, two button-joystick modules on the console, a telescope information display unit in the console, and a local button panel used to control the console processor. The system supports two subsystems which also provide console functions, the Programmable Video Digitizer System (PVDS), and the Panoramic Sky Monitor (PANSKY).

The button-joystick modules, the telescope display module, and the local button module communicate with the console processor through RS-232 asynchronous links running at 9600 baud. The PVDS and PANSKY subsystems communicate with the console processor through two separate shared memory systems. The console processor communicates with the Modcomp through the Direct Access Microprocessor Interface (DAMINT), a DMA type interface, and a 4805 controller on the Modcomp. Figure 1 is a $N^2$ diagram of the

hardware interface and Fig. 2 is a $N^2$ diagram of the data flow. To read the diagrams, first notice that different systems are placed in the boxes along the diagonal from upper left to lower right. To find out what data or hardware connection goes from one system to another, find the box which is horizontal from the sender and vertical from the receiver. In instances where a data flow exists but there is no hardware connection, the data has been routed through the console processor.

A dual system is implemented at the ETS. There are two telescopes, two Modcomp computers, and two consoles. Each console is associated with a particular telescope, but can be driven by either Modcomp. A block diagram of the console system is shown in Fig. 3.

B. Maintenance

Several aspects of the hardware, firmware, and software design facilitate maintainability of the system. The system is composed of several functional modules. Each module can be tested in isolation and can be exchanged between the two telescope systems. Where possible, commercially available products were used without modification. When custom-made boards were designed and built they were made sufficiently flexible that they could be used in several different modules for different purposes, with the firmware controlling their function in a particular system. This results in a small number of board types, so spares can be kept and field repair effected by board swap. Design, programming, and repair time were all reduced since personnel need only become familiar with a few board types.

Three types of intermodule communications links were used, RS-232, shared memory, and Modcomp DMA. All are serial links on a bit, byte, or

117490-N

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| MODCOMP | | | DAMINT | | | | |
| | PVDS | | SHARED MEMORY NO. 1 | | | | TV SIGNAL |
| | | PANSKY | SHARED MEMORY NO. 2 | | | | |
| DAMINT | SHARED MEMORY NO. 1 | SHARED MEMORY NO. 2 | CONSOLE PROCESSOR | RS-232 | RS-232 | RS-232 | |
| | | | RS-232 | BUTTON/ JOYSTICK | | | 5 TTL LINES |
| | | | RS-232 | | TELESCOPE DISPLAY | | 2 DRIVERS |
| | | | RS-232 | | | LOCAL BUTTON | |
| | TV SIGNAL | 16-CHANNEL A-D CONVERTER | | | | | OTHER SYSTEMS |

Fig. 1. Console system hardware interfaces

117491-N

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| MODCOMP | | DYNAMIC SCHEDULER DATA | LIGHTS, TELESCOPE INFORMATION | | | | | CRT |
| TRACKING PHOTOMETRY | PVDS | | LIGHTS | | | | | TV WITH CURSOR |
| | | PANSKY | A/D READINGS | | | | | ALL-SKY CAMERA VIDEO |
| BUTTONS, JOYSTICKS A/D | BUTTONS, JOYSTICKS | TELESCOPE DATA | CONSOLE PROCESSOR | LIGHTS | TELESCOPE INFORMATION, AUDIO ALARM | LIGHTS | | |
| | | | BUTTONS, JOYSTICKS | BUTTON/ JOYSTICK MODULE | | | VIDEO CHANNEL SELECT | LIGHTS |
| | | | | | TELESCOPE DISPLAY MODULE | | TORQUE MOTOR DIRECTION | TELESCOPE DATA, ALARM |
| | | | BUTTONS | | | LOCAL BUTTON MODULE | | ERROR AND TEST LIGHTS |
| | MAIN SENSOR TV | ANALOG INPUTS | | | | | OTHER SYSTEMS | |
| CRT | | | | BUTTONS, 2 JOYSTICKS | | SELECT MODCOMP, RUN TESTS | | OPERATOR |

Fig. 2. Console system data flow

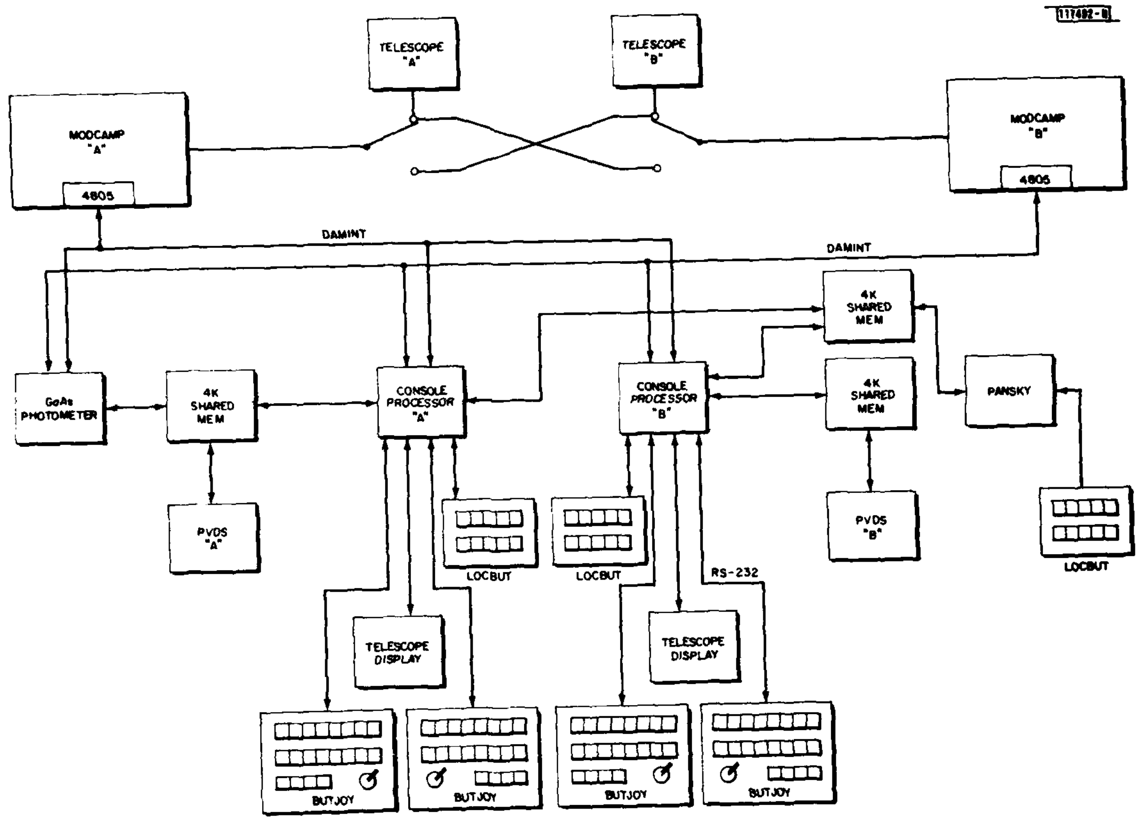


Fig. 3. Console interface systems

word basis. All are multiuse, in that many different types of information flow over the same hardware link. The advantage is that the link need only be tested for proper operation with one type of data, not with all types. The fewer number of hardware lines involved with serial links enormously reduces the number of electronic drivers and cables required. Since the remaining lines are more heavily used, faults are immediately apparent and easily isolated and repaired.

The use of microprocessors in each module allows very effective tests to be included. Tests are provided in each module to verify proper operation of that module, and to test the communications links between modules. A significant point is that the tests use special test programs, but not special test hardware. This avoids the problem of erroneous test results due to faulty test hardware. If proper operation is not verified, tests are provided to isolate the failure. Since the logic boards are standardized, the module can be repaired during operations by replacing the faulty board with a spare and inserting the appropriate program PROMs.

C. System Expansion

The console interface system was designed to accommodate further microprocessor based equipment. In general new equipment will wish to exchange information with the Modcomp or the console processor, or both. Five more microprocessors can be added to the Modcomp interface beyond the console processor (which uses less than 1% of the bandwidth) and a GaAs photometer currently planned. The interface handler on the Modcomp supports multiple task usage. The shared memory systems can be expanded to seven ports, while current plans only use three. Instruments associated with a

single console would share the PVDS shared memory system, and instruments which work with both consoles would share the PANSKY memory system. The console processor can support operation of these new instruments from the existing Button and Joystick modules, or a new instrument could have its own Local Button module, as PANSKY does.

Commercial instruments can be interfaced to a microprocessor to make them compatible with the console interface system. The microprocessors are much easier to interface than the Modcomp because they are single user systems without memory management, and a wide variety of compatible interface chips and boards are available.

## III. SYSTEM MODULES

### A. Console Processor

The Console Processor is a microcomputer based on the Motorola Micromodule series of microcomputer boards. It is the host processor by way of RS-232 serial links for two Button/Joystick (BUTJOY) modules, one Telescope Display (TELDIS) module, and one Local Button (LOCBUT) module. It communicates with either one of the two Modcomps through a high performance DMA channel. It communicates with the PVDS and PANSKY microcomputer based systems by way of two separate shared memory systems. The Console Processor is primarily concerned with passing data between the other processors and modules of the console interface system.

#### Hardware

The Console Processor hardware consists of six boards in a chassis with power supply. The boards are an M6809 based single board microcomputer, a PROM memory board, a four port serial interface board, the Direct Access Microprocessor Interface (DAMINT) board, a Shared Memory Master (SHAMEM) board, and a Shared Memory Client (CLIMEM) board.

The single board microcomputer board is an M68MM19 purchased from Motorola. It contains an M6809 microprocessor, running at 1 Mhz, 2 Kbytes of RAM, sockets for 8 Kbytes of PROM, a serial port, a timer, and a parallel port. In the Console Processor the serial port is configured as an RS-232 modem at 9600 baud and is used for diagnostics. The parallel port is not used. The timer is used to schedule periodic tasks. The onboard RAM is used for the system stack and temporary storage, and the program resides in PROM on the PROM board, which will hold up to 32K of 2716 type PROM.

The four port serial interface is a Motorola M68MM07 Quad Acia Module. The four ports are all configured as RS-232 modems at 9600 baud. The four ports are used to communicate with the left and right Button/Joystick modules, the Telescope Display module, and a Local Button module.

The DAMINT is a custom-made board which implements a high-speed DMA interface to either Modcomp computer. The board is connected to two Modcomp 4805 controllers, one on each Modcomp, which implement the DMA transfers on the Modcomp side. The board only talks to one Modcomp at a time under software selection. The board uses an MC6844 Direct Memory Access Controller (DMAC) chip to implement DMA functions in the microprocessor system. In addition, several custom hardware registers handle status and addressing lines. The DAMINT system allows the Modcomp to choose one of four channels in any of seven microcomputers (28 channels total) for data transfers. Transfer normally takes place at 200 Kbytes/second, with a 400 Kbytes/second mode also available. The Console Processor uses less than 1% of the bandwidth of the interface.

A custom-built shared memory system has been implemented to facilitate data transfers between Micromodule based systems. Two types of boards are used. The Shared Memory Master board contains 4 Kbytes of RAM, which is dual ported to the system containing the board and an external port. The Shared Memory Client board allows another system to access the Master board through the external port. Several Client boards can be daisy chained together, creating a multiple port memory system. At least seven ports are possible without violating timing restrictions.

Both a Master and a Client board are used in the console processor, but they belong to different shared memory systems. The Master board is

used to share a block of memory with the PVDS, which must get joystick and button information from the consoles, and returns light, centroid, and photometry information to the console processor and Modcomp. In fact, the data blocks sent to and from the Modcomp reside in this shared memory block, and are fully accessible to any system which has a port to the memory.

The Client board is part of a shared memory system which includes the other console processor and the Panoramic Sky Monitor (PANSKY), which contains the Master board. This memory system allows both consoles to access the Sky Monitor, and gives the Sky Monitor access to the positions of both telescopes.

A memory map of the console processor is given in Table 1 and a block diagram in Fig. 4. The program occupies 10 Kbytes and resides in 2716 type PROM on an M68MM04 PROM board.

Firmware

The primary function of the console processor is to provide a central point for information distribution. It receives data from various subsystems and processors, reformats the data, and sends it to other systems and processors over its various interfaces. Specific data formats are used for each interface (see Appendices B, C, D, E). It is the firmware which decodes these formats of incoming data and generates the formats of outgoing data.

Interrupt Driven Routines

The Modcomp interface and the serial interfaces to the BUTJOY and LOCBUT modules are interrupt driven. The task SWT running in the Modcomp communicates with the console processor, using channels 0 and 1 of the

TABLE I

MEMORY ASSIGNMENT IN CONSOLE PROCESSOR

| | |
|---|---|
| \$2ØØØ - \$4800 | Program PROM |
| \$8ØØØ - 8FFF | Master Shared Memory shared with PVDS |
| \$9ØØØ - 9FFF | Client Shared Memory port to PANSKY |
| \$EØØØ - E7FF | RAM on MM19 |
| \$EC1Ø - EC13 | PIA (unused) |
| \$EC14 - EC15 | Diagnostic ACIA |
| \$EC18 - EC1F | Timer |
| \$EC40 - EC5F | Modcomp Interface |
| \$ECAØ - ECA1 | ACIA port to left BUTJOY |
| \$ECA2 - ECA3 | ACIA port to TELDIS |
| \$ECA4 - ECA5 | ACIA port to right BUTJOY |
| \$ECA6 - ECA7 | ACIA port to LOCBUT |
| \$F000 - FFFF | PROM on MM19 (interrupt vectors) |

(\$ denotes hexadecimal (base 16) number)

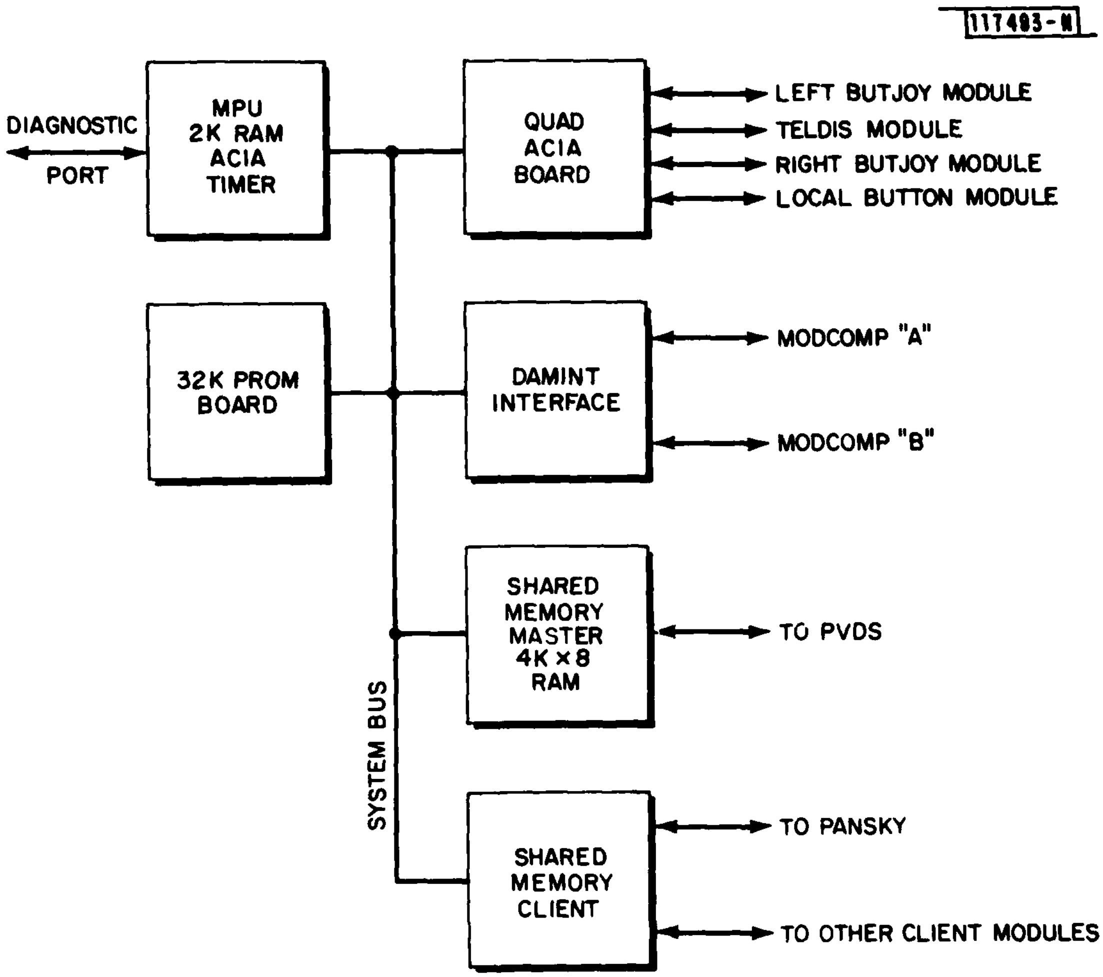


Fig. 4. Console processor

Modcomp interface. Channel 0 is a data block of 62 words (16 bits per word) read by the Modcomp. Channel 1 is a data block of 40 words sent by the Modcomp. At the end of these transfers an interrupt is generated by the DMAC controller chip. Any new light states are processed, buffers are cleared, and the channels reinitialized. Channel 2 of the Modcomp interface is not used at present. Channel 3 of the interface is used by the Dynamic Scheduler in the Modcomp to communicate with the Panoramic Sky Monitor. The channel is set to move 25 words from the Modcomp directly into the PANSKY shared memory. The console processor sets up the transfer and reinitializes the channel on the end of transfer interrupt, but does not handle the data. The data format of these common blocks are given in Appendix E.

Button depression reports from the two BUTJOY modules are placed in buffers for transmission to the Modcomp and PVDS. If the button Ø code is received, a light on the Local Button panel is toggled for testing purposes. Mixed in with button codes from the BUTJOY modules are the joystick information, sent as four pieces, 17 times a second. The console processor reassembles the pieces, scales them to the Modcomp format, and maintains current joystick positions in the Modcomp data block.

The buttons on the Local Button module are used to control the console processor, and are not reported to the Modcomp or PVDS. Buttons are provided to select either the 'A' or 'B' Modcomps, and to run communications checks to the BUTJOY, TELDIS, and LOCBUT modules.

A timer generates an interrupt 30 times a seconds. At this time the PVDS light buffer is checked for any light commands.

### Background Tasks

All time which is not spent servicing interrupts is spent in a loop which does extensive data reformatting. The information for the TELDIS display includes current GMT and sidereal times, the telescope commanded positions, and the telescope and dome actual positions. The information is received as 24 words, with three BCD digits packed into each word. The digits must be unpacked and sent one at a time to the TELDIS module. In addition, the information is converted back to binary format, and placed in the PANSKY shared memory. The Modcomp updates this information once a second, and the console processor waits for GMT seconds to change before starting the processing programs.

### Built-in Tests

Four buttons on the Local Button module are used to initiate and report on system level communications tests. A button is assigned to each of the TELDIS, LOCBUT, and two BUTJOY modules. Each button causes a $81 code to be sent to the associated module. The module responds by sending a $Ø (button Ø) code back. On receipt of a $Ø code, be it from pressing the test button on the module or in response to a $81 code, the button back-light state is toggled. This test shows that the module is alive and running its normal program, that the cabling is intact, and data can be sent in both directions. Each module has its own tests to check functions specific to that module.

A further level of tests can be accessed by connecting a diagnostic terminal to the console processor. The tests then available

allow individual lights on the modules to be controlled, the status of all interfaces to be displayed, and so on.

Tests of the Modcomp interface are available on the Modcomp.

B. Button/Joystick Module

The Button/Joystick (BUTJOY) module is the main operator input into the console interface system. It provides 50 push-button switches with computer controlled backlighting. It uses three of these buttons internally, leaving 47 to be defined by the host system. When a switch is depressed the event is reported to the host computer, which decides whether to backlight the button in response or take any other action. With this system a light change only occurs if the host has recognized the button pushed. The module also provides a two axis joystick, whose position is continuously reported to the host. The module also runs a video channel select subsystem, which allows the operator to choose a channel number, currently one of twenty, and reports this selection to the video distribution system.

The module size is 19" x 14.5" x 4.5", excluding connectors. The unit draws 2 amps of +5 volt power, 100 ma (max) of +15 and -15 power, and up to 2 amps of +24 volt power for the lamps.

Hardware

A system block diagram is given in Fig. 5. The module is controlled by an M6809 based microcomputer board known as the General Purpose Console Microcomputer (GPCM). The 50 switches are connected in a matrix run by an M6821 PIA. The 50 lights are driven by integrated circuit drivers connected to registers in the microcomputer system.

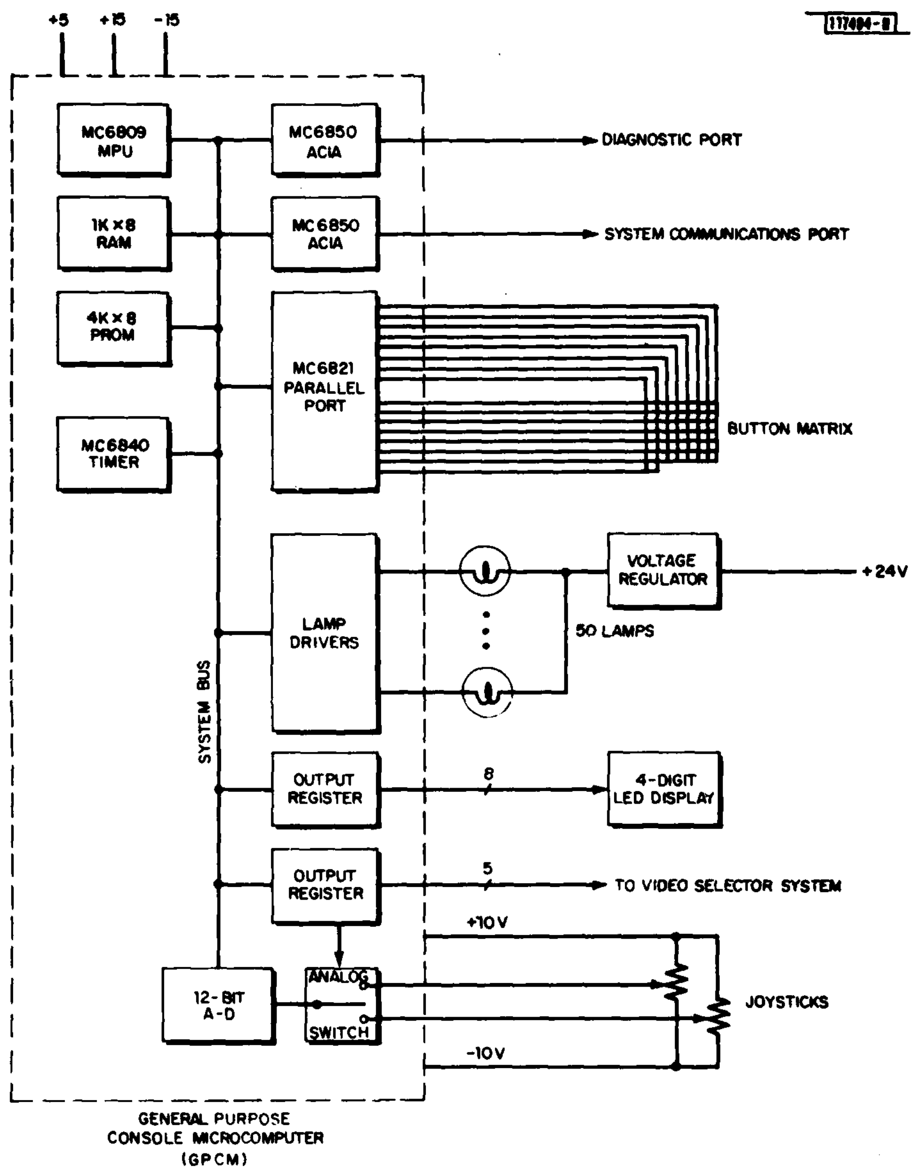


Fig. 5. Button/Joystick module

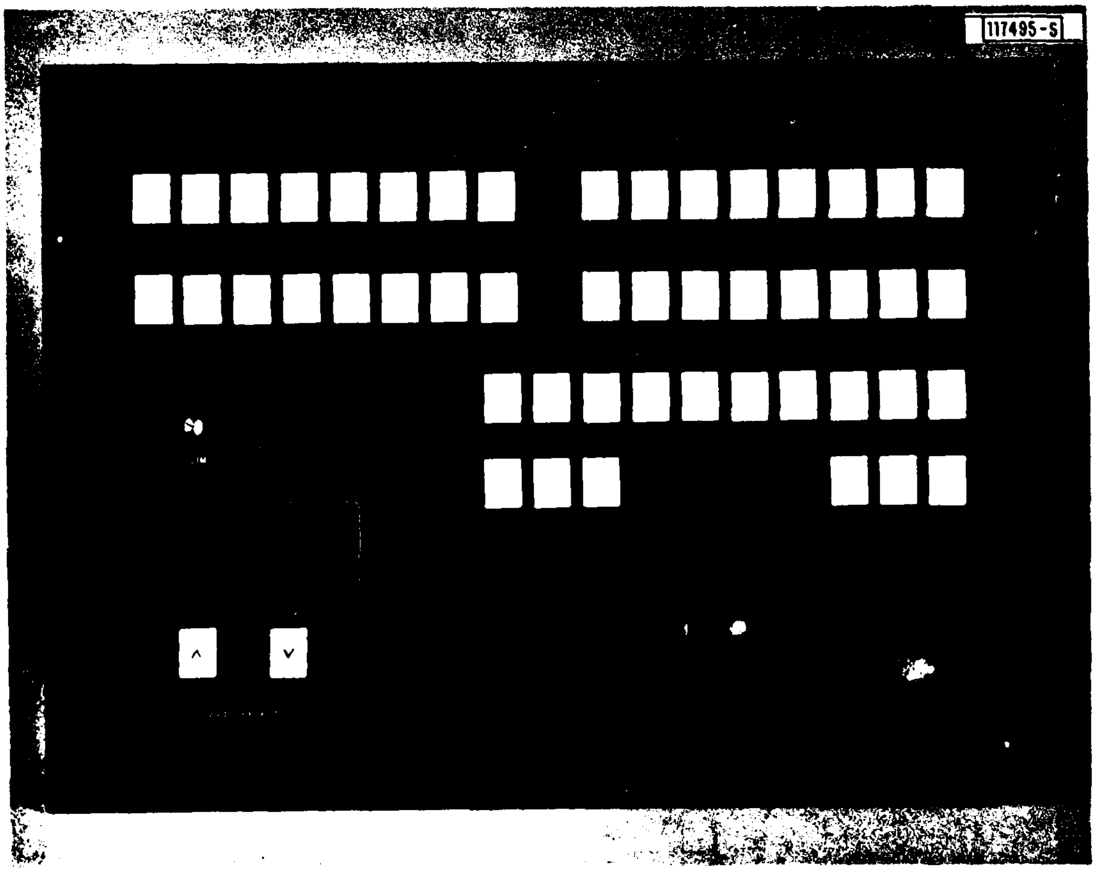


Fig. 6. Picture of BUTJOY

A four digit LED display is multiplexed to an 8-bit register, and the video channel select output is fed by another register. A 12-bit A-D converter and analog switch are used for reading the potentiometer type joystick.

A dimming control for the lights is provided by a voltage regulator controlled by a small potentiometer on the front panel.

Two serial communications ports are provided. One is wired as an RS-232 terminal and is normally used for system communications. The other is wired as an RS-232 modem and is normally only used for diagnostics. Both ports operate at 9600 baud. The memory map of the GPCM board is given in Appendix A.

Firmware

The microcomputer program determines how the unit works. After initialization the program is synchronized to a 200 hz timer, and repeats three steps, giving a program repetition rate of 67 hz.

The lights are never totally off. If they are commanded to be lit, they are on 100% of the time, but if commanded off they are dimly lit by operating them at a 33% duty cycle. This is accomplished by turning all lights on for the first step of the three-step program, and turning off those which are to be dim for the other two steps.

During the first program step all the button states are read by manipulating the switch matrix PIA. The current button states are compared to the states during the previous three program repetitions to perform a software debouncing function. New button depressions are

reported to the host computer over a serial communications link in the format described in Appendix B. No action is taken when a button is released.

During the second program step the video channel select subsystem is serviced and the joysticks are serviced. In the normal operating mode the currently selected channel is displayed in the middle two of the four digit LED display. The two buttons located beneath the display are used to increment or decrement the channel number. The number is changed when the button is depressed, and if the button is held down the number will continue to change at a fixed rate. The selected channel number is written out to the video distribution system.

The joystick is read every fourth time through the program loop. The analog switches are manipulated to perform an axis selection and sample and hold function. The joystick is read with a 12-bit A-D converter. The values are scaled, truncated to 10 bits, and sent to the host as four 5-bit pieces. One piece is sent every time through the program, so the complete joystick information is sent every 60 milliseconds.

During the third step of the program nothing is done. The lights remain in the state set at the beginning of the second step.

Communications from the host computer are handled on an interrupt basis. The incoming data is interpreted and acted upon according to the format described in Appendix A.

Built-in Tests

The BUTJOY contains many built-in tests. These are designed to verify proper operation and aid in isolating faults. Basic microcomputer

tests are run with the aid of a terminal through the diagnostic serial port, but the most commonly used tests are activated by depressing and holding button Ø at the middle left of the panel.

Since the lights are dimmed by a duty cycle and all lights are set to the dim state by a reset, the existence of any dimmed lights on the panel shows the program is running. The very fact that a light is dim shows that the lamp and its driver are fully functional.

When the test button is depressed several tests are run simultaneously. Depressing any other button will cause its light to toggle. This shows the button push was recognized. The upper six bits of the X and Y joystick position appear as two octal digits for eacn axis in the four digit LED display, and should change as the joystick is moved. The push of button 0 is reported to the host, and a smoothly increasing X and Y joystick position is also sent instead of the actual position.

These tests are run as long as the test button is held down, and normal operation is resumed as soon as the button is released.

C. Telescope Display Module

The Telescope Display Module (TELDIS) module is a module which displays in seven-segment LED numerals clock readings and telescope position and rate data. The information for the displays is obtained from the Modcomp computer and reformatted by the console processor, and is sent to the TELDIS module over an RS-232 serial communications line. Twelve parameters are displayed, with three to seven digits per parameter. A front panel push button activates built-in testing. In addition, the module contains an audio alarm and external device drivers.

The module size is 19" X 10.5" X 4.5", excluding connectors. The unit draws 5 amps of +5 volt power, and nominal amounts of +15 and -15 power for the RS-232 drivers.

Hardware

A system block diagram is shown in Fig. 7. The module is controlled by an M6809 based microcomputer which is a subset of the GPCM, and in fact a partially populated GPCM is used (described in Appendix A). The 70 LED digit displays are mounted on a single board. The LED devices used incorporate a data latch, decoder, and driver circuit, and need only be provided with a 4-bit data word and latch strobe. The microcomputer accesses the LEDs through a pair of registers, one of which generates the data and strobe, and the other which selects the LED to be written to. The data inputs of the LEDs are bused together, and address decoding circuitry on the LED board generates the individual latch strobes for the LEDs based on the address register contents.

The unit has a built-in audio alarm which is controlled by the Modcomp to warn of certain error conditions, such as the telescope being below its horizon limit. Sixteen general purpose driver circuits are available at a connector on the rear of the chassis. At present only two are used, and they control the direction of preload application on the telescope drives.

Two serial ports are provided. One is wired as an RS-232 terminal and is normally used for system communications. The other is wired as an RS-232 modem and is normally used only for diagnostic work. Both ports operate at 9600 baud.

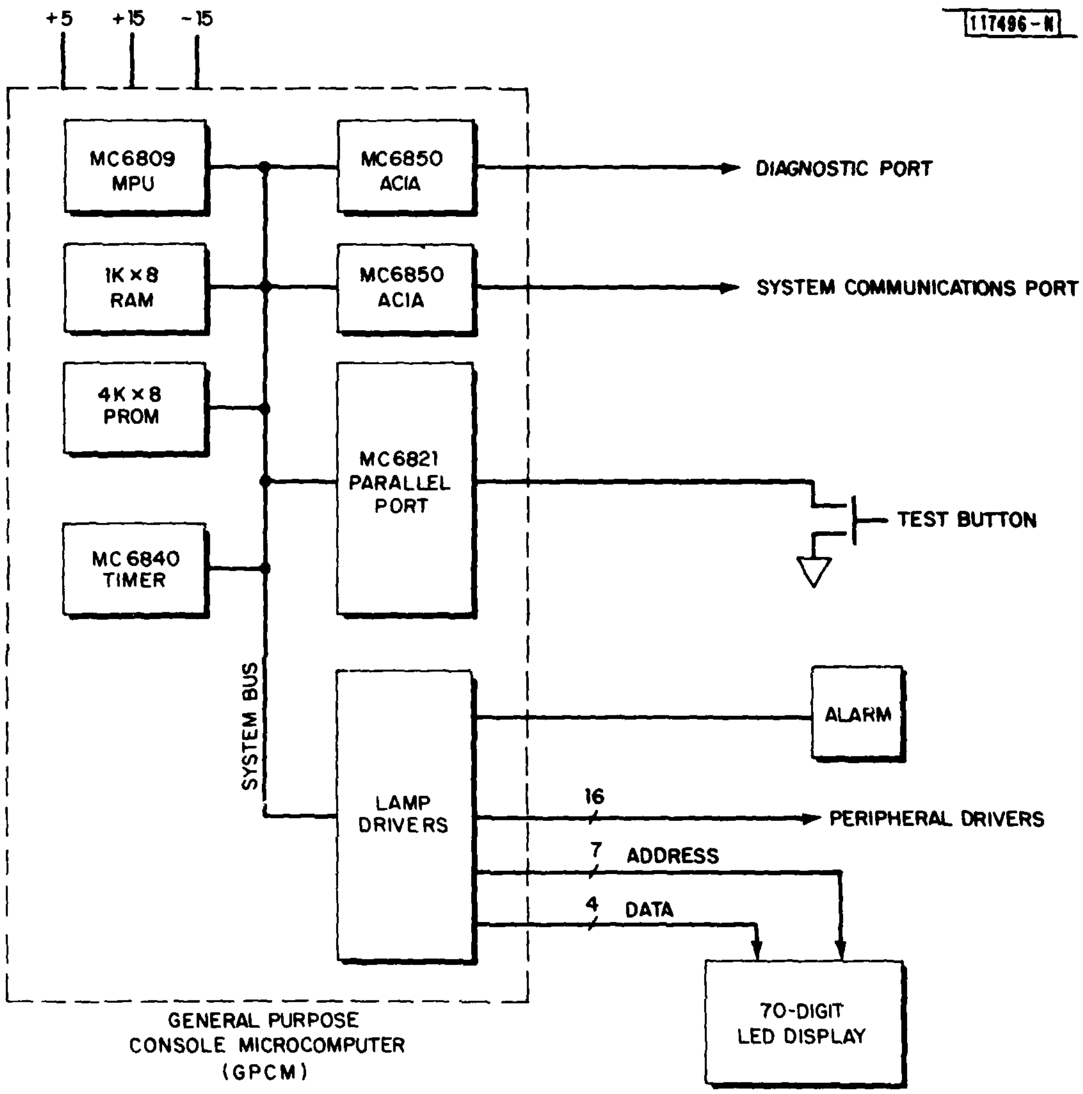


Fig. 7. Telescope display module

117497-S

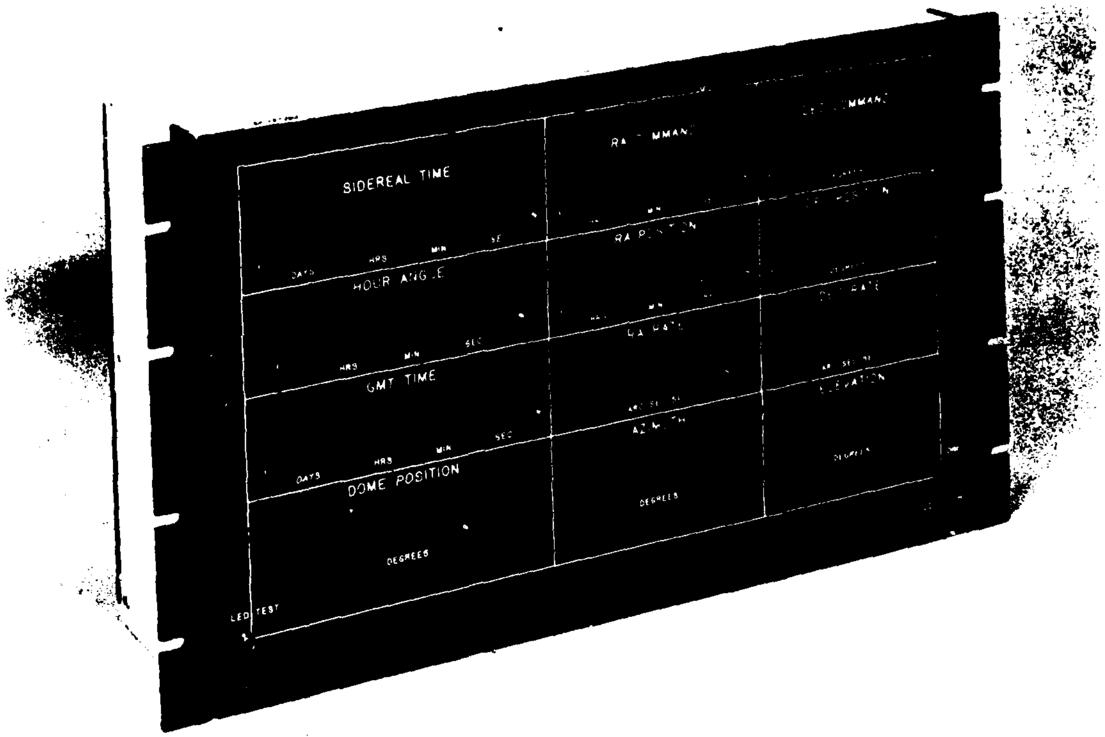


Fig. 8. Picture of TELDIS

Firmware

The microcomputer program determines how the unit works. The program spends most of its time waiting for a character to be received from the console processor and checking the test button. When a character is received, it is checked to see if it is a command, and if not the lower four bits are sent to the currently addressed LED and the LED address is advanced. Commands are provided to set the LED address to 0, to sound the audio alarm, to control the external driver circuits, to execute a communications check and to set all LEDs to '-'. (See Appendix C for communications format.)

When the test button is pushed, the hardware test line of the LED digits is grounded, lighting all segments. A $Ø data byte is also sent to the console processor. The audio alarm is sounded for roughly one second. When the test button is released, a test program is executed which sequentially writes the characters 0-9 to all LED digits.

D. Local Button Module

The Local Button (LOCBUT) module is a secondary input to the console interface system. It provides 32 pushbutton switches with host controlled backlighting. In addition it provides 16 peripheral driver lines at a rear panel connector, also host controlled. In the console interface system the module is used to select Modcomps, run systems communications tests, and indicate error conditions. The module occupies 5-1/4" of a standard 19" rack and is 11" deep.

Hardware

A system block diagram is given in Fig. 9. The module is controlled by an M6809 based microcomputer (a partially populated GPCM

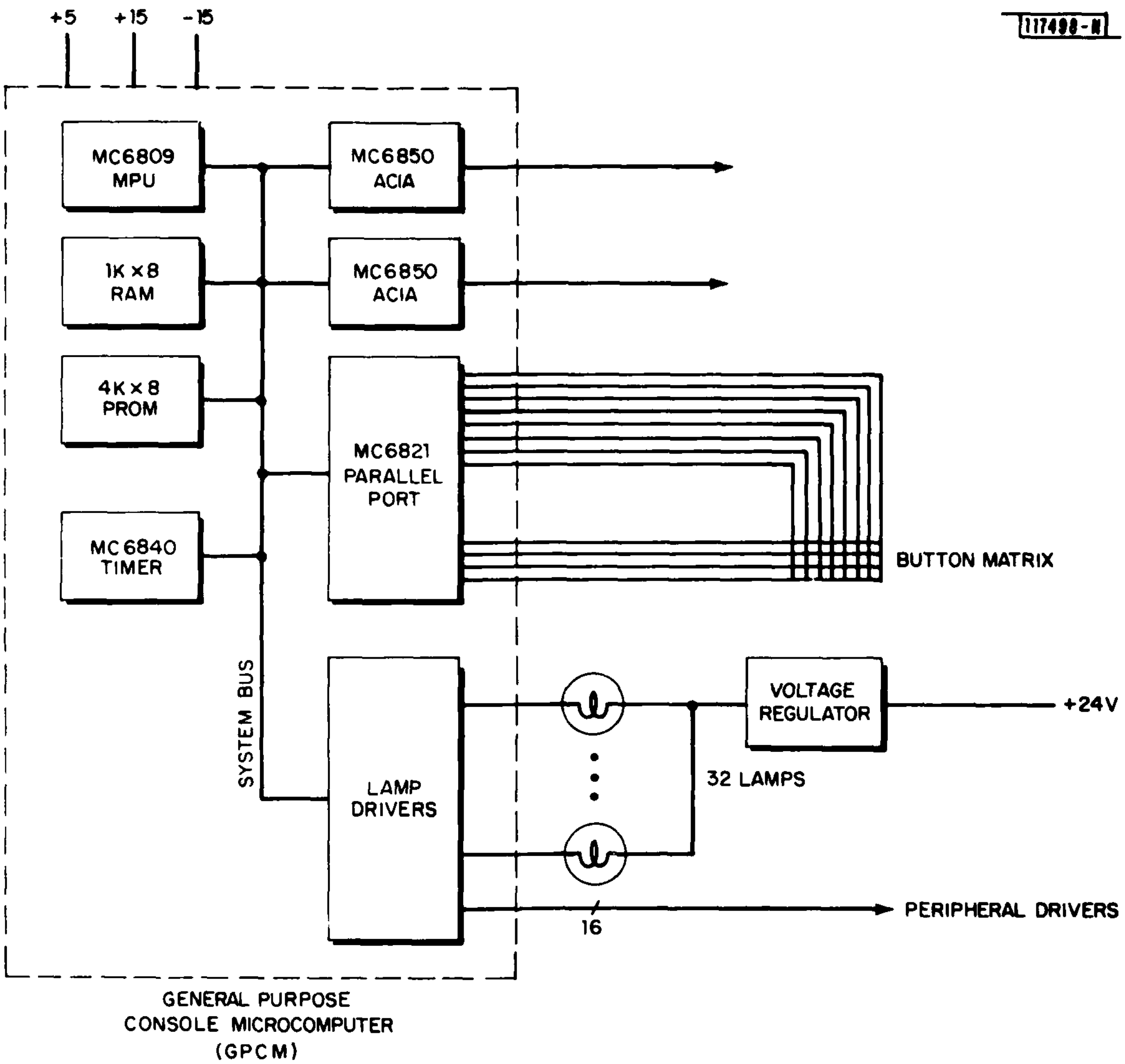


Fig. 9. Local button module

board is used). The button and light connections are very similar to the BUTJOY module, except that the light drivers for lights 32-47 are brought out to a rear panel connector.

A dimming control for the lights is provided by a voltage regulator controlled by a small potentiometer on the front panel.

Two serial communications ports are provided. One is wired as an RS-232 terminal and is normally used for system communications. The other is wired as an RS-232 modem and is normally only used for diagnostics. Both ports operate at 9600 baud. The memory map of the GPCM board is given in Appendix A.

<u>Firmware</u>

The microcomputer program runs the same three step light dimming scheme used in the BUTJOY program. However, the lamp drivers brought out to the rear panel are not cycled, but are simply on or off. In addition, rather than a light blinking state available in the BUTJOY modules, a light toggling command is available instead, which is better suited to indicate repeating test results or error occurrences. The communications formats are given in Appendix D.

When the test mode button at the upper left of the module is held down, depressing any other button will cause that light to toggle. The code for button zero will be sent when the test button is first depressed.

## E. <u>Programmable Video Digitizer System (PVDS)</u>

The PVDS is a microprocessor based system which digitizes television signals, processes the image, and generates cursor, alphanumeric, and gray scale overlays on the television monitors. In the context of

the console interface system, its most important function is the generation of a full screen crosshair which follows the joystick position of the BUTJOY module. It also generates object centroiding and photometry data which is passed on to the Modcomp through the console processor. The PVDS has access to button information and can control lights on the BUTJOY modules through memory shared with the console processor.

F. <u>Panoramic Sky Monitor (PANSKY)</u>

The Panoramic Sky Monitor is a microprocessor based system for evaluating local cloud cover for scheduling purposes. It includes a roof mounted television camera with a fisheye lens which covers the entire sky, a video processing and display unit, the microcomputer, and a private local button module. In the context of the console interface system it is unique in that it is the only system which shares memory with both console processors. It obtains the positions of the two telescopes from the console processors, and displays them on a video overlay. It includes a 16 channel, 12-bit A-D converter which is used to read weather information (temperature, wind speed and direction, and humidity), camera zoom settings, etc. This information is placed in shared memory and sent to both Modcomps.

The Panoramic Sky Monitor is also an output device of the Dynamic Scheduler program running in the Modcomp. It has its own channel on the DAMINT interface which places data directly in the shared memory. The console processor maintains the channel, but does not handle the data.

## IV. CURRENT STATUS

At this time both consoles at the ETS have been converted to the new system and are in daily use. A GaAs photometer is sharing the DAMINT interface to the Modcomp, and Panoramic Sky Monitor is interacting with the dynamic scheduler Modcomp program through the console processor. One PVDS is on line and providing centroid data for tracking experiments.

# APPENDIX A

## ADDRESS MAP OF GPCM

The upper two bits of the 16-bit address bus are not decoded, so all addresses are redundant in the blocks \$0000-\$3FFF, \$4000-\$7FFF, \$8000-\$BFFF and \$C000-\$FFFF.

| | |
|---|---|
| \$0000-\$03FF | RAM |
| \$2000-\$2007 | PTM Programmable Timer |
| \$2100-2105 | Lamp Drivers. Assignments vary and are given below. |
| \$2106-2107 | Write only output registers. Assignments vary with module and are given below. |
| \$2200-\$2203 | PIA Parallel Port. |
| \$2400-\$2401 | ACIA Diagnostic Port. |
| \$2500-\$2501 | ACIA Communications Port. |
| \$2700-\$2701 | A-D Converter (BUTJOY only). |
| \$3000-\$37FF | 2716 PROM (spare) |
| \$3800-\$3FFF | 2716 PROM |

All other addresses should be considered redundant and should not be accessed.

### GPCM DRIVER AND OUTPUT REGISTER USAGE

BUTJOY MODULE:

| | |
|---|---|
| \$2100-2105 | Lamp Drivers for lamp 0-47, with bit 7 of \$2100 being lamp 0 and bit 0 of \$2106 as lamp 47. A '1' in a bit position turns on that lamp. |

| | |
|---|---|
| \$2106 | Bits 7-4 are strobe bits for LEDs 0-3, respectively. A low value enables a write into that LED. Bits 0-3 are the LED data to be written. |
| \$2107 | Bit 5 controls lamps 48 and 49 (channel selectors). A '1' turns both on. Bit 6 connects the Y joystick to the A-D converter when low, and bit 5 connects the X joystick when low. Bits 0-4 are 5 bits (BCD) sent to the video distribution system as a channel select. |

TELDIS MODULE:

| | |
|---|---|
| \$2100 | Bits 0-6 are the LED address, in inverted form. |
| \$2101 | Bits 0-3 are the LED data, in inverted form. The data is written into the LED addressed by \$2100 by bringing bit 7 high, then low again. |
| \$2102-2103 | Peripheral drivers brought out to a rear connector. Bit 7 of \$2102 is driver 0, and bit 0 of \$2105 is driver 15. Setting a bit produces a low on the driver output which is an open collector output capable of sinking 300 ma at 80V. |
| \$2105 | Audio alarm driver. Writing \$FF into this location will turn on alarm, \$00 will turn it off. |

LOCBUT MODULE:

| | |
|---|---|
| \$2100-2103 | Lamp drivers for lamps 0-31, with bit 7 of \$2100 being lamp 0 and bit 0 of \$2103 being lamp 31. A '1' in a bit position will turn on that lamp. |

$2104-2105 Peripheral drivers brought out to rear connector. Bit 7 of $2102 is drive 0, and bit 0 of $2105 is driver 15. Setting a bit produces a low on the drive output, which is on open collector output capable of sinking 300 ma at 80V.

## APPENDIX B

### DATA FORMATS FOR BUTTON/JOYSTICK MODULE

Communications between the Button/Joystick (BUTJOY) module and a host are carried on over a 9600 baud RS-232 serial link. The asychronous data format is 8-bit data, no parity, 2-stop bits (MC6850 ACIA programming code $15). The format of the data is given below. Data from the BUTJOY module to the host tells which buttons have been pressed and what the current joystick position is. Data to BUTJOY controls the light states and there are a few special commands. The numbers above the word are the limiting bit positions, 0 being the LSB and 7 the MSB.

| 7 | 6 | 5 — 0 |
|---|---|---|
| Ø | Ø | BN |

<u>BUTTON REPORT FROM BUTJOY</u>

BN = Button number in the range 1-47

| 7 | 6 — 5 | 4 — 0 |
|---|---|---|
| 1 | JN | JD |

<u>JOYSTICK POSITION FROM BUTJOY</u>

JN = 0 JD is bits 9-5 of X Joystick (MSB)
1 JD is bits 4-0 of X Joystick (LSB)
2 JD is bits 9-5 of Y Joystick (MSB)
3 JD is bits 4-0 of Y Joystick (LSB)

JD = Joystick data

Note that a 10-bit joystick position is reported as a positive number with 00 being the upper left corner.

| 7 6 | 5 0 |
| --- | --- |
| M | LN |

LIGHT CONTROL FROM HOST

LN = Light number in range 0-47

M = ØØ: turn light off

01: turn light on

10: command mode

11: blink light at 2.5 hz

If command mode, then

LN = Ø: turn off all lights

1: send $ØØ back to host

## APPENDIX C

## DATA FORMATS FOR TELESCOPE DISPLAY MODULE

Communications between the Telescope Display (TELDIS) module and a host are carried on over a 9600 baud RS-232 serial link. The asynchronous data format is 8-bit data, no parity, 2-stop bits (MC6850 ACIA programming code \$15). Data is sent in a block, started by a \$8D character to clear a block pointer, and followed by 72 characters which are the LED digit information. The TELDIS module will send a \$00 character to the host when the test button is pushed or a \$81 is received as a communications check. The exact data formats are given below, and the position of each digit in the block are also given.

| 7 | 6 4 | 3 0 |
|---|---|---|
| Ø | | LD |

LED DIGIT DATA FROM HOST

| | |
|---|---|
| LD = 0-A: | place hex digit in next LED |
| B: | minus sign |
| C: | hex digit |
| D: | blank |
| E-F: | hex digit |

| 7 | 6 – 2 | 1 | 0 |
|---|---|---|---|
| Ø | | P | M |

+ - DIGIT DATA FROM HOST

M = 1: turn on -

0: turn off -

P = 1: turn on +

0: turn off +

| 7 | 6 | 5 – 0 |
|---|---|---|
| 1 | Ø | CN |

COMMANDS FROM HOST

CN = $0 clear all LEDs to '-'

= $1 send $ØØ back to host (communications check)

= $2 sound audio alarm for 0.2 seconds

= $D set LED digit pointer to Ø

| 7 – 0 | | | | | | |
|---|---|---|---|---|---|---|
| Ø | Ø | Ø | Ø | Ø | Ø | Ø |

TEST DATA TO HOST

Sent when the test button is pushed or a $81 is received.

| 7 | 6 | 5 | 4 | 3 | Ø |
|---|---|---|---|---|---|
| 1 | 1 | M | X | DN | |

<u>DRIVER COMMAND FROM HOST</u>

M = Ø    turn off driver DN

M = 1    turn on driver DN

DN =     driver number 0-15

POSITIONS OF DIGITS IN THE DATA BLOCK FOR TELDIS

| OFFSET | HEX | DIGIT |
|---|---|---|
| 0 | 0 | GMT 100 DAYS |
| 1 | 1 | " 10 " |
| 2 | 2 | " 1 " |
| 3 | 3 | " 10 HOURS |
| 4 | 4 | " 1 " |
| 5 | 5 | " 10 MINUTES |
| 6 | 6 | " 1 " |
| 7 | 7 | " 10 SECONDS |
| 8 | 8 | " 1 " |
| 9 | 9 | SID 100 DAYS |
| 10 | A | " 10 DAYS |
| 11 | B | " 1 " |
| 12 | C | " 10 HOURS |
| 13 | D | " 1 " |
| 14 | E | " 10 MINUTES |
| 15 | F | " 1 " |
| 16 | 10 | " 10 SECONDS |
| 17 | 11 | " 1 " |
| 18 | 12 | RA 10 HOURS |
| 19 | 13 | " 1 " |
| 20 | 14 | " 10 MINUTES |
| 21 | 15 | " 1 " |
| 22 | 16 | " 10 SECONDS |
| 23 | 17 | " 1 " |

| OFFSET | HEX | DIGIT |
|---|---|---|
| 24 | 18 | RA RATE + - |
| 25 | 19 | " 1000 ARC SEC/SEC |
| 26 | 1A | " 100 " |
| 27 | 1B | " 10 " |
| 28 | 1C | " 1 " |
| 29 | 1D | TEL AZ 100 DEGREES |
| 30 | 1E | " 10 " |
| 31 | 1F | " 1 " |
| 32 | 20 | " .1 " |
| 33 | 21 | DEC + - |
| 34 | 22 | " 10 DEGREES |
| 35 | 23 | " 1 " |
| 36 | 24 | " 10 MINUTES |
| 37 | 25 | " 1 " |
| 38 | 26 | " .1 " |
| 39 | 27 | DEC RATE + - |
| 40 | 28 | " 1000 ARC SEC/SEC |
| 41 | 29 | " 100 " |
| 42 | 2A | " 10 " |
| 43 | 2B | " 1 " |
| 44 | 2C | ELEVATION 100 DEGREES |
| 45 | 2D | " 10 " |
| 46 | 2E | " 1 " |
| 47 | 2F | UNUSED |

| OFFSET | HEX | DIGIT |
|---|---|---|
| 48 | 30 | HA + - |
| 49 | 31 | " 10 HOURS |
| 50 | 32 | " 1 " |
| 51 | 33 | " 10 MINUTES |
| 52 | 34 | " 1 " |
| 53 | 35 | " 10 SECONDS |
| 54 | 36 | " 1 " |
| 55 | 37 | DOME AZ 100 DEGREES |
| 56 | 38 | " 10 " |
| 57 | 39 | " 1 " |
| 58 | 3A | COMMANDED RA 10 HOURS |
| 59 | 3B | " 1 " |
| 60 | 3C | " 10 MINUTES |
| 61 | 3D | " 1 " |
| 62 | 3E | " 10 SECONDS |
| 63 | 3F | " 1 " |
| 64 | 40 | COMMANDED DEC + - |
| 65 | 41 | " 10 DEGREES |
| 66 | 42 | " 1 " |
| 67 | 43 | " 10 MINUTES |
| 68 | 44 | " 1 " |
| 69 | 45 | " 0.1 " |
| 70 | 46 | UNUSED |
| 71 | 47 | UNUSED |

## APPENDIX D

## DATA FORMATS FOR LOCAL BUTTON MODULE

Communications between the Local Button (LOCBUT) module and a host are carried on over a 9600 baud RS-232 serial link. The asynchronous data format is 8-bit data, no parity, 2 stop bits (MC6850 ACIA programming code $15). When a button is pressed, the button number is sent to host. Data is sent from the host to control button lighting, control 16 outputs on a rear panel connector, and execute a communications check. The data formats are given below:

| 7 6 | 5 0 |
|---|---|
| M | LN |

LIGHT CONTROL FROM HOST

LN = Light/driver number. Lights are 0-31, drivers are 32-47.

M = 00: Turn light/driver off.

01: Turn light/driver on.

10: Command mode.

11: Toggle light/driver.

If command mode,

LN = 0: Clear all lights/drivers.

1: Send $00 back to host.

| 7 | | 5 | 4 | 0 |
|---|---|---|---|---|
| Ø | Ø | Ø | BN | |

BUTTON REPORT FROM LOCBUT

BN = Button number in range 1-31

## APPENDIX E

## CONSOLE PROCESSOR COMMON BLOCK STRUCTURE

The console processor has two shared memory systems, each 4K bytes in size. The first provides a global common for the console processor, PVDS, and other systems associated with a single console. The second is shared by both consoles and the Panoramic Sky Monitor (PANSKY). The contents of these shared memories are given below, with positions denoted by an offset from the base address of the shared memory. The base address can be set to any 4K boundary in each subsystem.

Shared Memory #1. Located at \$8000-\$8FFF in the console processor.

| OFFSETS | |
|---|---|
| \$0000-\$0015 | Console Processor control variables. |
| \$0000 | MCOMP - Image of Modcomp select register of DAMINT interface. |
| \$0016-\$0091 | Data block sent to Modcomp, 124 bytes. |
| \$16 | Left Joystick, X coordinate in joystick format. |
| \$18 | Left Joystick, Y coordinate. |
| \$1A | Right Joystick, X coordinate. |
| \$1C | Right Joystick, Y coordinate. |
| \$1E | Button buffer of 8 two byte values in button format. |
| \$2E | Readings of housekeeping A-D converter, 16 two byte values in A-D units. |
| \$4E | PVDS centroid, X coordinate in joystick format. |
| \$50 | PVDS centroid, Y coordinate. |
| \$52-\$91 | Unused, but sent. |

| | |
|---|---|
| $0092-$0097 | Console Processor control variables. |
| $0098-$00E7 | Data block received from Modcomp, 80 bytes. |
| $0098 | TELDIS information, 24 two byte words. |
| $00C8 | Light commands, buffer of 16 two byte words, in lights format. |
| $00E8-$011B | PVDS Communications area. |
| $00EE | Button buffer of 10 two byte values in button format. |
| $0108 | Light buffer of 10 two byte values in light format. |
| $011C-$0FFF | Miscellaneous and spare. |

| 15 | 12 | 11 0 |
|---|---|---|
| Ø Ø Ø Ø | | JY |

JOYSTICK FORMAT (16 BIT)

JY = Joystick value. $ØØØ is upper edge or left side, $FFF is lower edge or right side.

| 15 8 | 7 0 |
|---|---|
| Ø Ø Ø Ø Ø Ø Ø Ø | BN |

BUTTON FORMAT (16 BIT)

BN = Button number. Ø is not allowed and denotes an empty buffer location. 1-47 are the left BUTJOY module and 64-111 are the right BUTJOY module.

| 15 | 14 10 | 9 8 | 7 0 |
|---|---|---|---|
| C | Ø Ø Ø Ø Ø | M | LN |

<u>LIGHTS FORMAT (16 BIT)</u>

LN = Light number. 1-47 are the left BUTJOY module, 48-63 are drivers on the TELDIS module, 64-111 are on the right BUTJOY module.

M = ØØ Turn light off.

= 01 Turn light on.

= 10 Not used.

= 11 Put light in blinking state

C = 1 Special command. Only one special command is currently defined, and that is C = 1 and LN = $80 will clear all modules.

*Shared Memory #2*. Located at $9000-$9FFF in the console processor.

| OFFSETS | |
|---|---|
| $0000-$001F | Raw readings of housekeeping A-D converter. |
| $0100-$01FF | A console information for PANSKY. See below for content. |
| $0200-$02FF | B console information for PANSKY. See below for content. |
| $0300-$0FFF | Unused. |

Console Information for PANSKY. Within each console information block are the following data at the offsets indicated. All values are 16 bit integers.

| | |
|---|---|
| \$00 | GMT DAYS |
| \$02 | " HOURS |
| \$04 | " MINUTES |
| \$06 | " SECONDS |
| \$08 | SIDEREAL DAYS |
| \$0A | " HOURS |
| \$0C | " MINUTES |
| \$0E | " SECONDS |
| \$10 | RIGHT ASCENSION HOURS (current position) |
| \$12 | " " MINUTES |
| \$14 | " " SECONDS |
| \$16 | RIGHT ASCENSION RATE (10 X $\widehat{\text{sec}}$/sec) |
| \$18 | AZIMUTH (to nearest degree) |
| \$1A | DECLINATION DEGREES (current position) |
| \$1C | " MINUTES (to nearest) |
| \$1E | DECLINATION RATE (10 X $\widehat{\text{sec}}$/sec) |
| \$20 | ELEVATION (to nearest degree) |
| \$22 | HOUR ANGLE HOURS |
| \$24 | " " MINUTES |
| \$26 | " " SECONDS |
| \$28 | DOME POSITION (AZIMUTH in degrees) |

| | |
|---|---|
| $2A | RIGHT ASCENSION HOURS (commanded position) |
| $2C | " " MINUTES |
| $2E | " " SECONDS |
| $30 | DECLINATION DEGREES (commanded position) |
| $32 | " MINUTES (to nearest) |

Dynamic Scheduler Information for PANSKY. Within each console information block is the following data at the offset indicated.

| | |
|---|---|
| $CD | Number of valid position pairs to follow (16 bit). |
| $DØ-$FF | Positions of satellites as a 16 bit azimuth in degrees followed by a 16 bit elevation in degrees. |

## ACKNOWLEDGMENTS

Many people worked on this system at one time or another. In particular, modules were built by 5 different technicians and significant programming done by 5 different people. It would be very difficult to single some out for special recognition without slighting the others, so I will just say it was a group effort.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER<br>ESD-TR-82-048 | 2. GOVT ACCESSION NO.<br>AD-A119156 | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>New Console Interface System | | 5. TYPE OF REPORT & PERIOD COVERED<br>Project Report |
| | | 6. PERFORMING ORG. REPORT NUMBER<br>Project Report ETS-65 |
| 7. AUTHOR(s)<br>Andrew J. Wardrop | | 8. CONTRACT OR GRANT NUMBER(s)<br>F19628-80-C-0002 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Lincoln Laboratory, M.I.T.<br>P.O. Box 73<br>Lexington, MA 02173-0073 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>Program Element Nos. 63428F and 12424F<br>Project Nos. 2698 and 2295 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Air Force Systems Command, USAF<br>Andrews AFB<br>Washington, DC 20331 | | 12. REPORT DATE<br>14 July 1982 |
| | | 13. NUMBER OF PAGES<br>54 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)<br>Electronic Systems Division<br>Hanscom AFB, MA 01731 | | 15. SECURITY CLASS. (of this report)<br>Unclassified |
| | | 15a. DECLASSIFICATION DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

| | | |
|---|---|---|
| GEODSS | console | shared memory |
| ETS | computer interface | microprocessor |

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A new console interface system has been designed and built for the Experimental Test Site (ETS) of the Ground Electro-Optical Deep Space Surveillance (GEODSS) program using microprocessor technology. The new system allows current and future microprocessor based subsystems to be conveniently integrated into the overall system. Serial interfaces, high performance, multi-use DMA interfaces, and shared memory systems are extensively used. The use of LSI, standard microcomputer boards, and built-in tests and diagnostics is expected to result in a more reliable and maintainable system.

DD FORM 1473 1 Jan 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

LME

8